



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/807,374	03/24/2004	Yutaka Ota	250813US2	3964
22850	7590	08/10/2007	EXAMINER	
OBLON, SPIVAK, MCCLELLAND, MAIER & NEUSTADT, P.C.			QIAN, SONGWEI	
1940 DUKE STREET			ART UNIT	PAPER NUMBER
ALEXANDRIA, VA 22314			2109	
NOTIFICATION DATE		DELIVERY MODE		
08/10/2007		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patentdocket@oblon.com  
oblonpat@oblon.com  
jgardner@oblon.com

<b>Office Action Summary</b>	Application No.	Applicant(s)	
	10/807,374	OTA, YUTAKA	
	Examiner Songwei Qian	Art Unit 2109	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) Responsive to communication(s) filed on 03/24/2004.
- 2a) This action is FINAL.                    2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) Claim(s) 1-20 is/are pending in the application:
  - 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-20 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 24 March 2004 is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) All    b) Some \* c) None of:
    1. Certified copies of the priority documents have been received.
    2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
    3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 03/24/2004 and 05/30/2006
- 4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) Notice of Informal Patent Application
- 6) Other: \_\_\_\_\_

***DETAILED ACTION***

1. Claims 1-20 are pending in this application.

***Claim Rejections - 35 USC § 101***

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-12 and 20 are rejected under 35 U.S.C. 101 as the claimed invention is directed to non-statutory subject matter.

4. In claims 1-12, a "compiler" is being recited; however, it appears that the compiler would reasonably be interpreted by one of ordinary skill in the art as software, per se. Software, per se, is not one of statutory subject matter.

5. In claim 20, a "tool" is being recited; however, it appears that the tool would reasonably be interpreted by one of ordinary skill in the art as software, per se. Software, per se, is not one of statutory subject matter.

***Claim Rejections - 35 USC § 112***

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 1, 13, and 20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

8. As for claims 1, 13, and 20, the phrase "in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database" renders the claims indefinite because it is unclear as to what this phrase means and how "agreement" is measured.

9. As for claims 7-10 and 17, the phrase "plural definitions of details of the processing operations" renders the claims indefinite because it is unclear as to what this phrase means.

***Claim Rejections - 35 USC § 103***

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashida (US Pub. # 2002/0019973 A1) in view of Santhanam et al. (US Pat. # 6,247,174 B1), hereinafter "Santhanam".

12. As for claim 1, Hayashida discloses:

A compiler for generating object code from an input source program ([0157], lines 11-12), the compiler comprising:

a syntax analyzer configured to analyze whether or not an operation described in the source program conforms to grammatical rules, and to analyze whether or not a combination of the operations defines an intrinsic function and details of processing operations of the intrinsic function ([0157], lines 13-20);

an intrinsic function definition database configured to store a definition of the intrinsic function and the details of the processing operations of the intrinsic function ([0157], lines 21-24), as analyzed by the syntax analyzer (FIG. 1);

a code generator configured to generate machine instructions from the source program based on a result of the processing of the syntax analyzer ([0157], lines 25-29

and FIG. 1); and

However, Hayashida does not explicit disclose:

a code optimizer configured to optimize the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function, in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database.

On the other hand, Santhanam discloses:

a code optimizer (607, FIG. 6) configured to optimize the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function (inline.h, FIG.6), in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database (FIG. 3 and 6, Col. 11, lines 59-62, and Col. 15, lines 24-31; note that 301 in FIG 3 is the intrinsic function definition database).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Hayashida with the teachings of Santhanam by configuring a code optimizer to optimize the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic

function, in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database in order to enable the routines to be optimized more effectively and thereby improve run-time application performance.

13. As for claim 13, Hayashida discloses:

A computer implemented method of compiling for generating object code from an input source program ([0157], lines 11-12), the computer implemented method comprising:

analyzing, by a syntax analyzer, whether or not an operation described in the source program conforms to grammatical rules, and analyzing, by the syntax analyzer, whether or not a combination of the operations defines an intrinsic function and details of processing operations of the intrinsic function ([0157], lines 13-20);

storing a definition of the intrinsic function and the details of the processing operations of the intrinsic function, as analyzed by the syntax analyzer in an intrinsic function definition database ([0157], lines 21-24 and FIG. 1);

generating, by a code generator, machine instructions from the source program based on a result of the processing of the syntax analyzer([0157], lines 25-29 and FIG. 1);

However, Hayashida does not explicit disclose:

optimizing, by a code optimizer, the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function, in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database.

On the other hand, Santhanam discloses:

optimizing, by a code optimizer (607, FIG. 6), the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function (inline.h, FIG.6), in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database (FIG. 3 and 6, Col. 11, lines 59-62, and Col. 15, lines 24-31; note that 301 in FIG 3 is the intrinsic function definition database).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Hayashida with the teachings of Santhanam by optimizing, by a code optimizer, the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function, in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored

in the intrinsic function definition database in order to enable the routines to be optimized more effectively and thereby improve run-time application performance.

14. As for claims 2 and 14, Hayashida discloses:

a lexical analyzer configured to divide the operations described in the source program into tokens ([0157], lines 13-14), wherein the syntax analyzer analyzes whether or not the tokens conforms to grammatical rules, and analyzes whether or not the combination of the tokens defines the intrinsic function and the details of the processing operations of the intrinsic function ([0157], lines 15-20).

15. As for claims 3 and 15, Hayashida discloses:

the syntax analyzer (12, FIG. 1) inputs the definition of the intrinsic function and the details of the processing operations of the intrinsic function from an intrinsic function information file (18, FIG. 1) different from the source program (1a, FIG.1).

16. As for claims 4-6, Hayashida discloses:

the definition of the intrinsic function includes information of parameter types and an identification name (S32 and S33, FIG. 3 and [0157], lines 60-62).

17. As for claims 7-10 and 17, Hayashida discloses:

plural definitions of details of the processing operations (instruction attribute information, [0157], lines 22-23) can be defined in the intrinsic function definition database relative to a single intrinsic function ([0157], lines 21-24).

18. As for claim 11, Hayashida discloses:

the definition of the intrinsic function and the details of the processing operations of the intrinsic function ([0157], lines 21-24) can be described by C language ([0002]).

19. As for claim 12, Hayashida discloses:

the definition of the intrinsic function and the details of the processing operations of the intrinsic function ([0157], lines 21-24) can be described by hardware description language ([0002]; note that hardware description language is high-level language):

20. As for claim 16, Hayashida discloses:

the definition of the intrinsic function including information of parameter types and an identification name are analyzed by the syntax analyzer (11, FIG. 1) and stored in the intrinsic function definition database (18, FIG. 1).

21. As for claim 18, Hayashida discloses:

the definition of the details of the processing operations of the intrinsic function ([0157], lines 21-24) described by C language ([0002]) is analyzed by the syntax

analyzer (12, FIG. 1).

22. As for claim 19, Hayashida discloses:

the definition of the intrinsic function and the details of the processing operations of the intrinsic function ([0157], lines 21-24) described by hardware description language ([0002]; note that hardware description language is high-level language) is analyzed by the syntax analyzer (12, FIG. 1).

23. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Hayashida in view of Santhanam, and further in view of Curreri et al.(US Pat. # 6,091,896), hereinafter “Curreri”.

24. As for claim 20, Hayashida discloses:

A program development tool for designing an application program for a processor installed user-defined hardware ([0157], lines 11-12), the program development tool comprising:

a compiler for generating object code from the application program ([0157], lines 11-12) comprising

a lexical analyzer configured to divide an operation described in the application program into tokens ([0157], lines 13-14),

a syntax analyzer configured to analyze whether or not the tokens conform to grammatical rules, and to input a hardware definition of a user-defined instruction and

convert the input hardware definition into a definition of the intrinsic function and details of processing operations of the intrinsic function ([0157], lines 15-20),

an intrinsic function definition database configured to store the definition of the intrinsic function and the details of the processing operations of the intrinsic function, converted by the syntax analyzer ([0157], lines 21-24), as analyzed by the syntax analyzer (FIG. 1),

a code generator configured to generate machine instructions from the application program based on a result of the processing of the lexical analyzer and a result of processing of the syntax analyzer ([0157], lines 25-29 and FIG. 1),

However, Hayashida does not explicit disclose:

a code optimizer configured to optimize the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function, in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database; and

a simulator configured to debug the application program compiled by the compiler.

On the other hand, Santhanam discloses:

a code optimizer (607, FIG. 6) configured to optimize the machine instructions to machine instructions corresponding to the details of the processing operations of the

intrinsic function (inline.h, FIG.6), in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database (FIG. 3 and 6, Col. 11, lines 59-62, and Col. 15, lines 24-31; note that 301 in FIG 3 is the intrinsic function definition database);

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Hayashida with the teachings of Santhanam by configuring a code optimizer to optimize the machine instructions to machine instructions corresponding to the details of the processing operations of the intrinsic function, in a case where a string of the machine instructions generated by the code generator are in agreement with the details of the processing operations of the intrinsic function stored in the intrinsic function definition database in order to enable the routines to be optimized more effectively and thereby improve run-time application performance.

But neither Hayashida nor Santhanam explicit discloses:

a simulator configured to debug the application program compiled by the compiler.

However, Curreri discloses:

a simulator (104, FIG. 1) configured to debug the application program (106, FIG. 1) compiled by the compiler (102, FIG. 1).

It would have been obvious to one of ordinary skill in the art at the time of invention was made to combine the teachings of Hayashida and Santhanam with the teachings of Curreri by having a simulator configured to debug the application program compiled by the compiler in order to be able to debug optimized code (Curreri, Col. 1, lines 17).

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Songwei Qian whose telephone number is 571-270-1910. The examiner can normally be reached on M-F (alternative Friday off 8:00am thru 5:00pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nabil El-Hady can be reached on 571-272-3963. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

SQ  
08/02/2007

*N. Elwady*  
NADIA M ELWADY  
SUPERVISORY PATENT EXAMINER